TrackMe

LUCA GRELLA 905655
DANIELE LUNGHI 905083

SOFTWARE ENGINEERING 2 PROJECT PRESENTATION

GOALS

- Application Specific goals
  - ✦ Data4Help:
    - G4. Allow Third Part User to access the data of Normal User, upon acceptance
    - G5. Allow TPU to access anonymous data of a group of at least one thousand people
  - ✦ AutomatedSOS:
    - G6. Notify ill if health values are below threshold for more than 5 seconds
- Common goals
  - G1. Normal User's account correct handling
  - G2. Third Part User's account correct handling
  - G3. Users data registration

THE WORLD AND THE MACHINE: AN EXAMPLE

WORLD

- User wants to access data of a group of people

SHARED

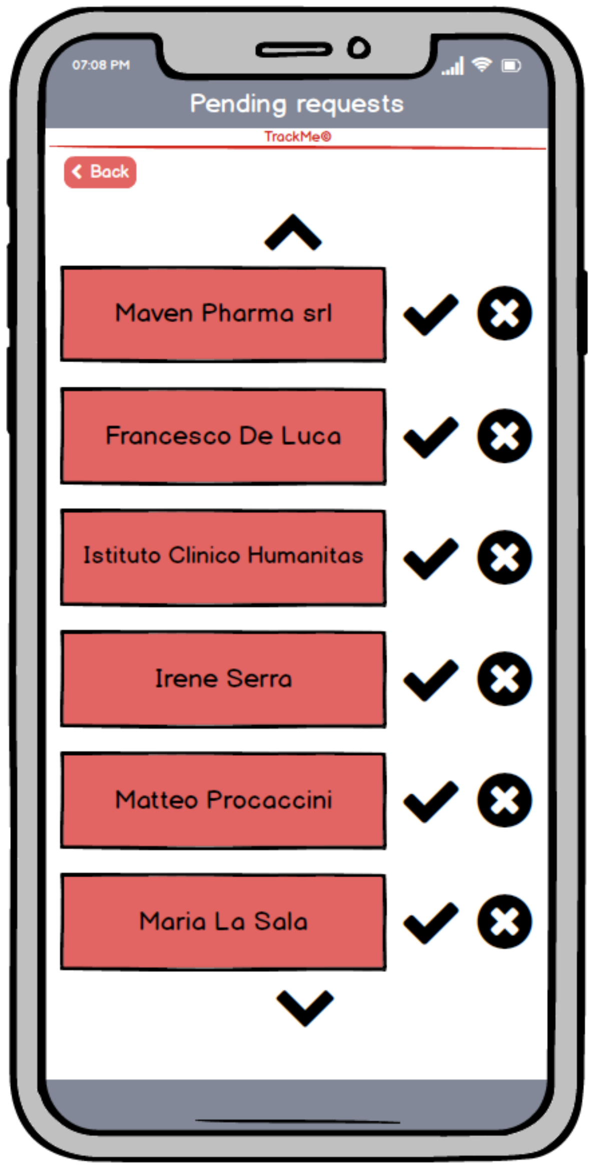
- User requests anonymized data of a group of individuals
- Data are shown to the user

MACHINE

- Query execution

A MEANINGFUL USE CASE

USE CASE	**CONFIRM REQUEST**
GOALS	G1
ACTOR	NU
ENTRY CONDITIONS	NU successfully logged in and he received a request from a TPU to access his data.
EVENTS FLOW	1. NU clicks on “Requests” red button. 2. He checks the pending requests. 3. He provided with the option of accepting the request and let the TPU access his data pressing the “V” button, or denying the request pressing the “X” button.
EXIT CONDITIONS	Request is now handled, and the third part user receives the answer.
EXCEPTIONS	1. RequestNotFoundException 2. GenericException

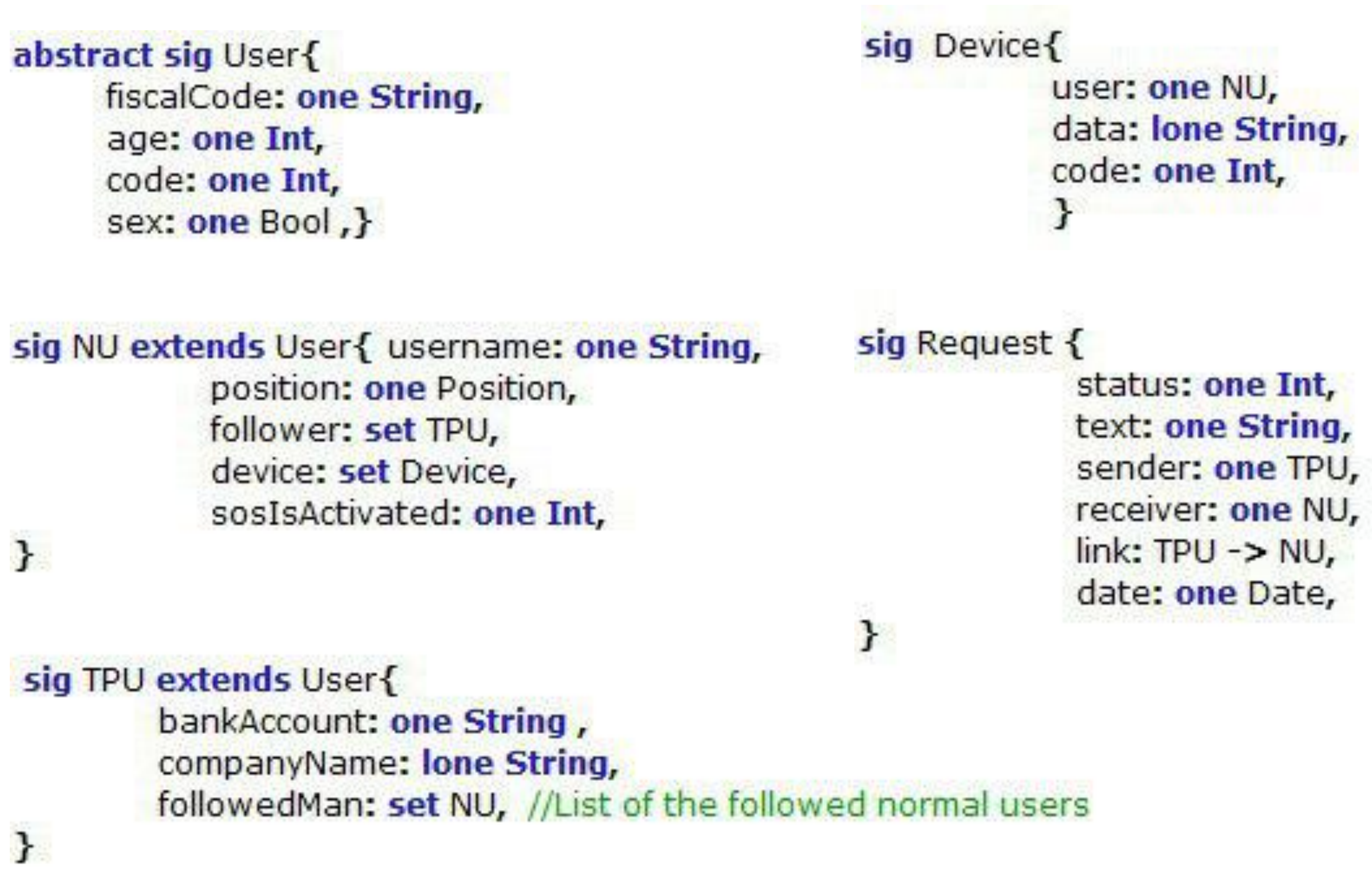


ALLOY WORLD

- Signatures

We model the main classes which are part of our system. In particular we create two types of user who are connect by means of request.

We gave a particular focus on the relationship among a normal user and his devices.



```
abstract sig User{
        fiscalCode: one String,
        age: one Int,
        code: one Int,
        sex: one Bool ,}

sig NU extends User{ username: one String,
            position: one Position,
            follower: set TPU,
            device: set Device,
            sosIsActivated: one Int,
}

sig TPU extends User{
        bankAccount: one String ,
        companyName: lone String,
        followedMan: set NU,  //List of the followed normal users
}
```

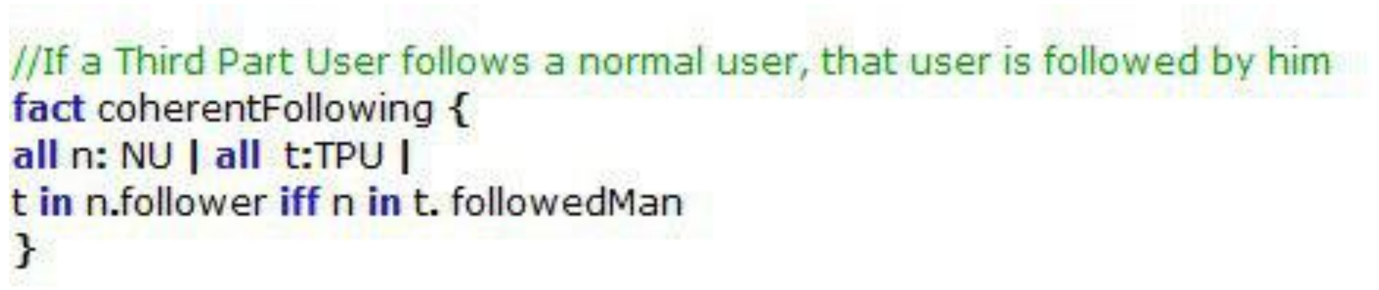
```
sig  Device{
            user: one NU,
            data: lone String,
            code: one Int,
            }

sig Request {
            status: one Int,
            text: one String,
            sender: one TPU,
            receiver: one NU,
            link: TPU -> NU,
            date: one Date,
}
```

ALLOY WORLD

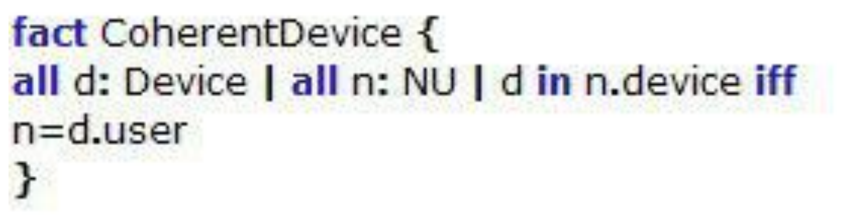
- Facts

To be a follower, a third part user must belong to the followers list of the individual user.



```
//If a Third Part User follows a normal user, that user is followed by him
fact coherentFollowing {
all n: NU | all  t:TPU |
t in n.follower iff n in t. followedMan
}
```

The device ownership relationship is symmetric.



```
fact CoherentDevice {
all d: Device | all n: NU | d in n.device iff
n=d.user
}
```

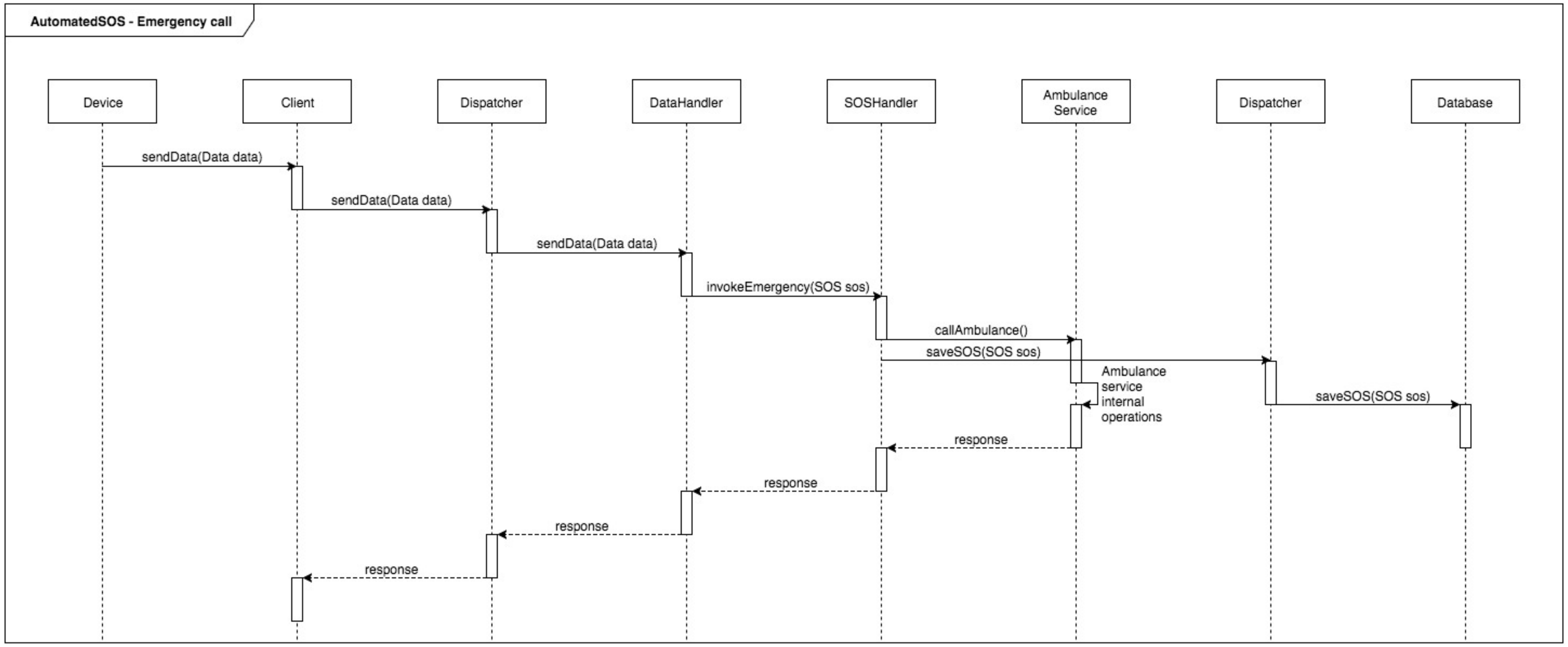
DD THROUGH SOS

LOGIC AND INTRODUCTION

- System is complex, AutomatedSOS is simple
- The main elements of the system are involved in the process: highly representative!
- But how does it work?

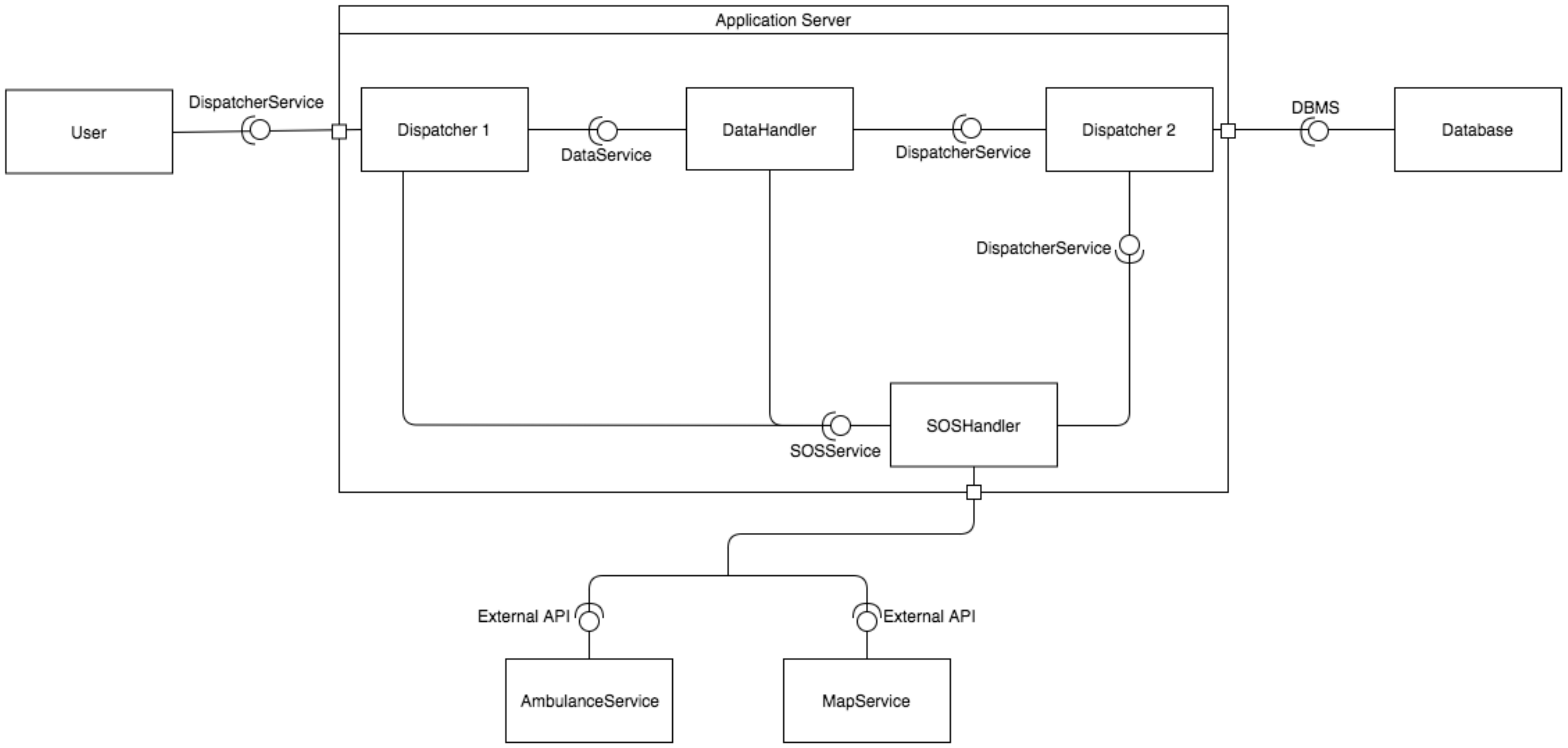
DD THROUGH SOS

RUNTIME VIEW



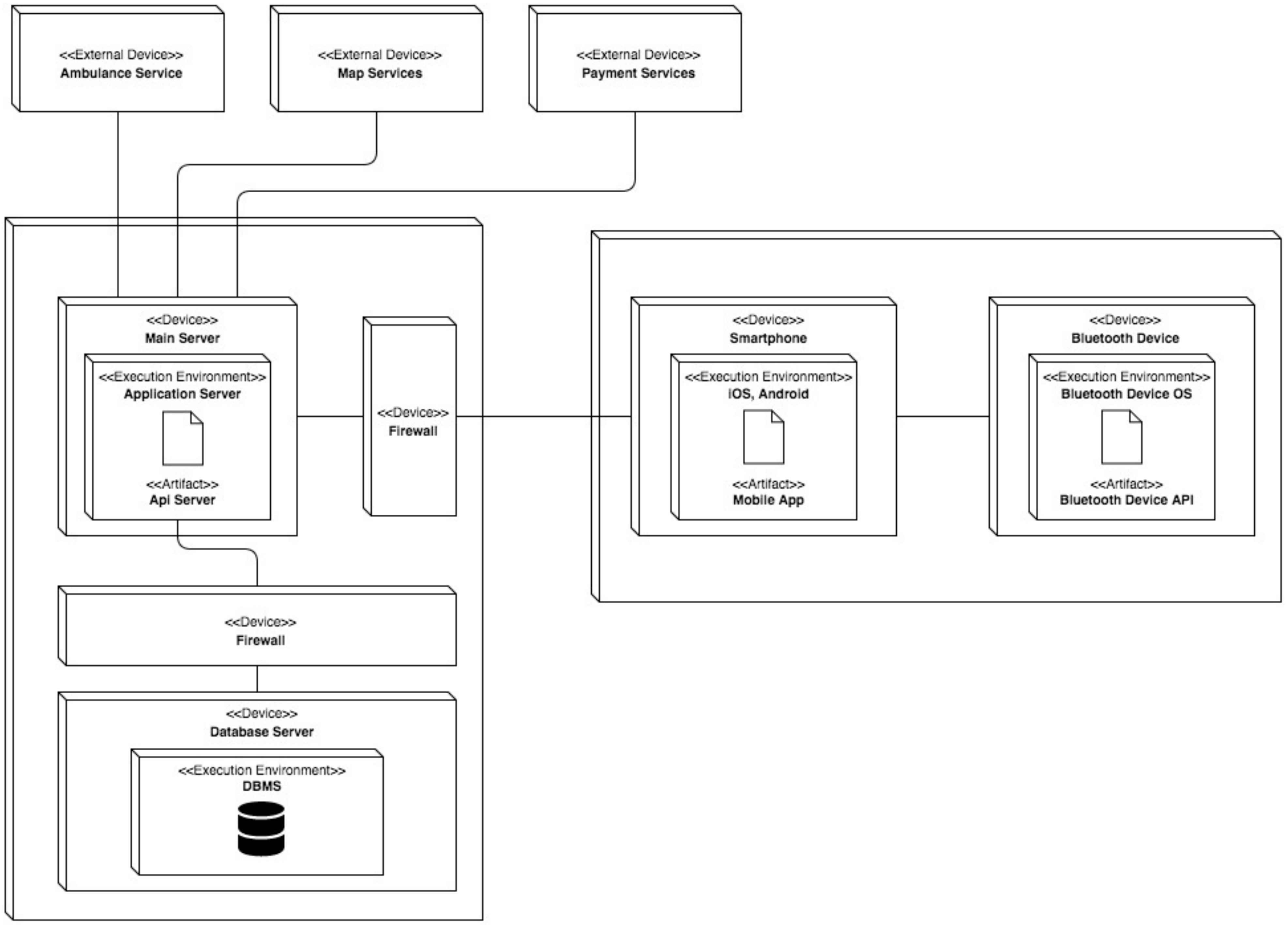
DD THROUGH SOS

COMPONENTS AND DEPLOYMENT 1



DD THROUGH SOS

COMPONENTS AND DEPLOYMENT 2

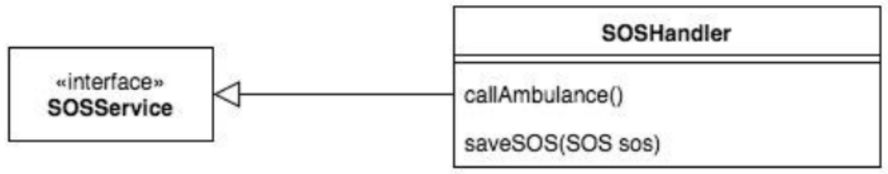


DD THROUGH SOS

INTERFACES AND INTEGRATION

- Interfaces

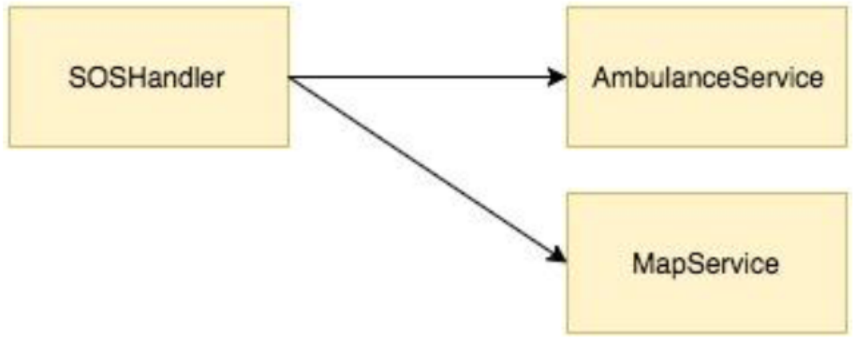
All the components implement one interface



- Integration

SOSHandler interacts with some external components, by means of the necessary APIs

Integration must be tested in parallel with the implementation of the single components



DESIGN DECISIONS AND PATTERNS

- COTS solution for the Data Base
- MVC
- Client-Server
- Observer
- Façade

THANKS FOR YOUR ATTENTION!

Luca Grella and Daniele Lunghi